

Negotiation Games

Philipp Hoffmann

Fakultät für Informatik
Technische Universität München
Germany
ph.hoffmann@tum.de

Negotiations, a model of concurrency with multi party negotiation as primitive, have been recently introduced in [5, 6]. We initiate the study of games for this model. We study *coalition problems*: can a given coalition of agents force that a negotiation terminates (resp. block the negotiation so that it goes on forever)?; can the coalition force a given outcome of the negotiation? We show that for arbitrary negotiations the problems are EXPTIME-complete. Then we show that for sound and deterministic or even weakly deterministic negotiations the problems can be solved in PTIME. Notice that the input of the problems is a negotiation, which can be exponentially more compact than its state space.

1 Introduction

In [5, 6], the first author and Jörg Desel have introduced a model of concurrency with multi party negotiation as primitive. The model allows one to describe distributed negotiations obtained by combining “atomic” multi party negotiations, or *atoms*. Each atom has a number of *parties* (the subset of agents involved), and a set of possible outcomes. The parties agree on an outcome, which determines for each party the subset of atoms it is ready to engage in next.

Ill-designed negotiations may deadlock, or may contains useless atoms, i.e., atoms that can never be executed. The problem whether a negotiation is well designed or *sound* was studied in [5, 6]. The main result was the identification of two classes, called deterministic and acyclic weakly deterministic negotiations, for which the soundness problem is tractable: while the problem is PSPACE-complete for arbitrary negotiations, it becomes polynomial for these two classes.

In this paper we start the study of games on negotiations. As for games played on pushdown automata [12], vector addition systems with states (VASS) [3], counter machines [9], or asynchronous automata [10], games on negotiations can be translated into games played on the (reachable part of the) state space. However, the number of states of a negotiation may grow exponentially in the number of agents, and so the state space can be exponentially larger than the negotiation. We explore the complexity of solving games *in the size of the negotiation*, not on the size of the state space. In particular, we are interested in finding negotiation classes for which the winner can be decided in polynomial time, thus solving the state space explosion problem.

We study games formalizing the two most interesting questions related to a negotiation. First, can a given *coalition* (i.e., a given subset of agents) force termination of the negotiation? (Negotiations may contain cycles.) Second, can the coalition force a given final outcome?

Our first results show that these two problems are EXPTIME-complete in the size of the negotiation. This is the case even if the negotiation is deterministic, and so it seems as if the tractability results of [5, 6] cannot be extended to games. But then, we are able to show that, very surprisingly, the problems are polynomial for deterministic (or even weakly deterministic) negotiations *that are sound*. This is very

satisfactory: since unsound negotiations are ill-designed, we are not interested in them anyway. And, very unexpectedly, the restriction to sound negotiations has as collateral effect a dramatic improvement in the complexity of the problem. Moreover, the restriction comes “at no cost”, because deciding soundness of deterministic negotiations is also decidable in polynomial time.

The full version of this paper including the appendix is available on arXiv.org.

Related work. Our games can be seen as special cases of concurrent games [2, 1] in which the arena is succinctly represented as a negotiation. Explicit construction of the arena and application of the algorithms of [2, 1] yields an exponential algorithm, while we provide a polynomial one.

Negotiations have the same expressive power as 1-safe Petri nets or 1-safe VASS, although they can be exponentially more compact (see [5, 6]). Games for unrestricted VASS have been studied in [3]. However, in [3] the emphasis is on VASS with an infinite state space, while we concentrate on the 1-safe case.

The papers closer to ours are those studying games on asynchronous automata (see e.g. [10, 7, 8]). Like negotiations, asynchronous automata are a model of distributed computation with a finite state space. These papers study algorithms for deciding the existence of distributed strategies for a game, i.e., local strategies for each agent based only on the information the agent has on the global system. Our results identify a special case with much lower complexity than the general one, in which local strategies are even memoryless.

Finally, economists have studied mathematical models of negotiation games, but with different goals and techniques (see e.g. [11]). In our terminology, they typically consider negotiations in which all agents participate in all atomic negotiations. We focus on distributed negotiations, where in particular atomic negotiations involving disjoint sets of agents may occur concurrently.

2 Negotiations: Syntax and Semantics

Negotiations are introduced in [5]. We recall the main definitions. We fix a finite set A of *agents* representing potential parties of negotiations. In [5, 6] each agent has an associated set of internal states. For the purpose of this paper the internal states are irrelevant, and so we omit them.

Atoms. A *negotiation atom*, or just an *atom*, is a pair $n = (P_n, R_n)$, where $P_n \subseteq A$ is a nonempty set of *parties*, and R_n is a finite, nonempty set of *outcomes*.

(Distributed) Negotiations. A distributed negotiation is a set of atoms together with a *transition function* \mathcal{X} that assigns to every triple (n, a, r) consisting of an atom n , a party a of n , and an outcome r of n a set $\mathcal{X}(n, a, r)$ of atoms. Intuitively, this is the set of atomic negotiations agent a is ready to engage in after the atom n , if the outcome of n is r .

Formally, given a finite set of atoms N , let $T(N)$ denote the set of triples (n, a, r) such that $n \in N$, $a \in P_n$, and $r \in R_n$. A *negotiation* is a tuple $\mathcal{N} = (N, n_0, n_f, \mathcal{X})$, where $n_0, n_f \in N$ are the *initial* and *final* atoms, and $\mathcal{X} : T(N) \rightarrow 2^N$ is the *transition function*. Further, \mathcal{N} satisfies the following properties: (1) every agent of A participates in both n_0 and n_f ; (2) for every $(n, a, r) \in T(N)$: $\mathcal{X}(n, a, r) = \emptyset$ iff $n = n_f$.

Graphical representation. Negotiations are graphically represented as shown in Figure 1. For each atom $n \in N$ we draw a black bar; for each party a of P_n we draw a white circle on the bar, called a *port*. For each $(n, a, r) \in T(N)$, we draw a hyper-arc leading from the port of a in n to all the ports of a in the atoms of $\mathcal{X}(n, a, r)$, and label it by r . Figure 1 shows two Father-Daughter-Mother negotiations. On the left, Daughter and Father negotiate with possible outcomes yes (y), no (n), and ask_mother (am). If the outcome is the latter, then Daughter and Mother negotiate with outcomes yes, no. In the negotiation on

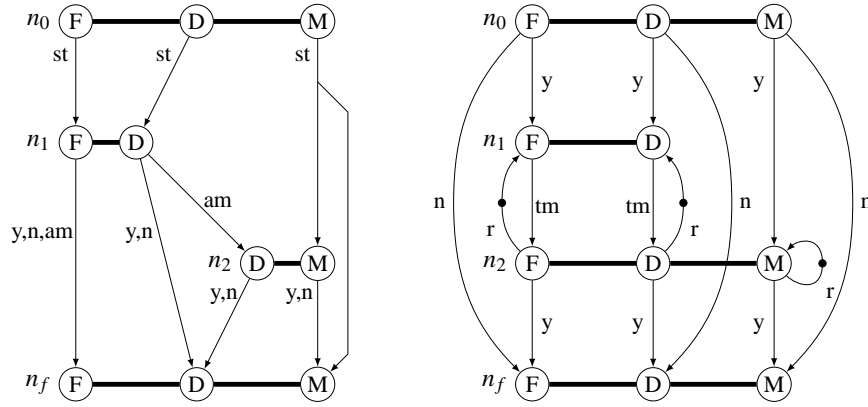


Figure 1: An acyclic and a cyclic negotiation.

the right, Father, Daughter and Mother negotiate with outcomes yes and no. If the outcome is yes, then Father and Daughter negotiate a return time (atom n_1) and propose it to Mother (atom n_2). If Mother approves (outcome yes), then the negotiation terminates, otherwise (outcome r) Daughter and Father renegotiate the return time.

Semantics. A *marking* of a negotiation $\mathcal{N} = (N, n_0, n_f, \mathcal{X})$ is a mapping $\mathbf{x}: A \rightarrow 2^N$. Intuitively, $\mathbf{x}(a)$ is the set of atoms that agent a is currently ready to engage in next. The *initial* and *final* markings, denoted by \mathbf{x}_0 and \mathbf{x}_f respectively, are given by $\mathbf{x}_0(a) = \{n_0\}$ and $\mathbf{x}_f(a) = \emptyset$ for every $a \in A$.

A marking \mathbf{x} *enables* an atom n if $n \in \mathbf{x}(a)$ for every $a \in P_n$, i.e., if every party of n is currently ready to engage in it. If \mathbf{x} enables n , then n can take place and its parties agree on an outcome r ; we say that (n, r) *occurs*. Abusing language, we will call this pair also an outcome. The occurrence of (n, r) produces a next marking \mathbf{x}' given by $\mathbf{x}'(a) = \mathcal{X}(n, a, r)$ for every $a \in P_n$, and $\mathbf{x}'(a) = \mathbf{x}(a)$ for every $a \in A \setminus P_n$. We write $\mathbf{x} \xrightarrow{(n,r)} \mathbf{x}'$ to denote this.

By this definition, $\mathbf{x}(a)$ is always either $\{n_0\}$ or equals $\mathcal{X}(n, a, r)$ for some atom n and outcome r . The marking \mathbf{x}_f can only be reached by the occurrence of (n_f, r) (r being a possible outcome of n_f), and it does not enable any atom. Any other marking that does not enable any atom is a *deadlock*.

Reachable markings are graphically represented by placing tokens (black dots) on the forking points of the hyper-arcs (or in the middle of an arc). Figure 1 shows on the right a marking in which F and D are ready to engaging n_1 and M is ready to engage in n_2 .

We write $\mathbf{x}_1 \xrightarrow{\sigma} \mathbf{x}_k$ to denote that there is a sequence

$$\mathbf{x}_1 \xrightarrow{(n_1, r_1)} \mathbf{x}_2 \xrightarrow{(n_2, r_2)} \dots \xrightarrow{(n_{k-1}, r_{k-1})} \mathbf{x}_k \xrightarrow{(n_k, r_k)} \mathbf{x}_{k+1} \dots$$

such that $\sigma = (n_1, r_1) \dots (n_k, r_k) \dots$. If $\mathbf{x}_1 \xrightarrow{\sigma}$, then σ is an *occurrence sequence* from the marking \mathbf{x}_1 , and \mathbf{x}_1 enables σ . If σ is finite, then we write $\mathbf{x}_1 \xrightarrow{\sigma} \mathbf{x}_{k+1}$ and say that \mathbf{x}_{k+1} is *reachable* from \mathbf{x}_1 .

Soundness. A negotiation is *sound* if (a) every atom is enabled at some reachable marking, and (b) every occurrence sequence from the initial marking either leads to the final marking \mathbf{x}_f , or can be extended to an occurrence sequence that leads to \mathbf{x}_f .

The negotiations of Figure 1 are sound. However, if we set in the left negotiation $\mathcal{X}(n_0, M, st) = \{n_2\}$ instead of $\mathcal{X}(n_0, M, st) = \{n_2, n_f\}$, then the occurrence sequence $(n_0, st)(n_1, yes)$ leads to a deadlock.

Determinism and weak determinism. An agent $a \in A$ is *deterministic* if for every $(n, a, r) \in T(N)$ such that $n \neq n_f$ there exists an atom n' such that $\mathcal{X}(n, a, r) = \{n'\}$.

The negotiation \mathcal{N} is *weakly deterministic* if for every $(n, a, r) \in T(N)$ there is a deterministic agent b that is a party of every atom in $\mathcal{X}(n, a, r)$, i.e., $b \in P_{n'}$ for every $n' \in \mathcal{X}(n, a, r)$. In particular, every reachable atom has a deterministic party. It is *deterministic* if all its agents are deterministic.

Graphically, an agent a is deterministic if no proper hyper-arc leaves any port of a , and a negotiation is deterministic if there are no proper hyper-arcs. The negotiation on the left of Figure 1 is not deterministic (it contains a proper hyper-arc for Mother), while the one on the right is deterministic.

3 Games on Negotiations

We study a setting that includes, as a special case, the questions about coalitions mentioned in the introduction: Can a given coalition (subset of agents) force termination of the negotiation? Can the coalition force a given concluding outcome?

In many negotiations, there are reachable markings that enable more than one atom. If two of those atoms share an agent, the occurrence of one might disable the other and they are not truly concurrent. For a game where we want to allow concurrent moves, we formalize this concept with the notion of an independent set of atoms.

Definition 3.1. A set of atoms S is *independent* if no two distinct atoms of S share an agent, i.e., $P_n \cap P_{n'} = \emptyset$ for every $n, n' \in S$, $n \neq n'$.

It follows immediately from the semantics that if a marking \mathbf{x} enables all atoms of S and we fix an outcome \mathbf{r}_i for each $n_i \in S$, then there is a unique marking \mathbf{x}' such that $\mathbf{x} \xrightarrow{\sigma} \mathbf{x}'$ for every sequence $\sigma = (n_1, \mathbf{r}_1) \dots (n_k, \mathbf{r}_k)$ such that each atom of S appears exactly once in σ . In other words, \mathbf{x}' depends only on the outcomes of the atoms, and not on the order in which they occur.

A *negotiation arena* is a negotiation whose set N of atoms is partitioned into two sets N_1 and N_2 . We consider concurrent games [2, 1] with three players called Player 1, Player 2, and Scheduler. At each step, Scheduler chooses a nonempty set of independent atoms among the atoms enabled at the current marking of the negotiation arena. Then, Player 1 and Player 2, independently of each other, select an outcome for each atom in $S \cap N_1$ and $S \cap N_2$, respectively. Finally, these outcomes occur in any order, and the game moves to the unique marking \mathbf{x}' mentioned above. The game terminates if it reaches a marking enabling no atoms, otherwise it continues forever.

Formally, a partial play is a sequence of tuples $(S_i, F_{i,1}, F_{i,2})$ where each $S_i \subseteq N$ is a set of independent atoms and $F_{i,j}$ assigns to every $n \in S_i \cap N_j$ an outcome $\mathbf{r} \in R_n$. Furthermore it must hold that every atom $n \in S_i$ is enabled after all atoms in S_0, \dots, S_{i-1} have occurred with the outcomes specified by $F_{0,1}, F_{0,2}, \dots, F_{i-1,1}, F_{i-1,2}$. A play is a partial play that is either infinite or reaches a marking enabling no atoms. For a play π we denote by π_i the partial play consisting of the first i tuples of π .

We consider two different winning conditions. In the *termination game*, Player 1 wins a play if the play ends with n_f occurring, otherwise Player 2 wins. In the *concluding-outcome game*, we select for each agent a a set of outcomes G_a such that $n_f \in \mathcal{X}(n, a, r)$ for $r \in G_a$ (that is, after any outcome $\mathbf{r} \in G_a$, agent a is ready to terminate). Player 1 wins if the the play ends with n_f occurring, and for each agent a the last outcome (n, \mathbf{r}) of the play such that a is a party of n belongs to G_a .

A strategy σ for Player j , $j \in \{1, 2\}$ is a partial function that, given a partial play $\pi = (S_0, F_{0,1}, F_{0,2}), \dots, (S_i, F_{i,1}, F_{i,2})$ and a set of atoms S_{i+1} returns a function $F_{i+1,j}$ according to the constraints above. A play π is said to be played according to a strategy σ of Player j if for all i , $\sigma(\pi_i, S_{i+1}) = F_{i+1,j}$. A strategy σ is a winning strategy for Player j if he wins every play that is played according to σ . Player j is said to win the game if he has a winning strategy. Notice that if Player 1 has a winning strategy then he wins every play against any pair of strategies for Player 2 and Scheduler.

Definition 3.2. Let \mathcal{N} be a negotiation arena. The termination (resp. concluding-outcome) problem for \mathcal{N} consists of deciding whether Player 1 has a winning strategy for the termination game (concluding-outcome game).

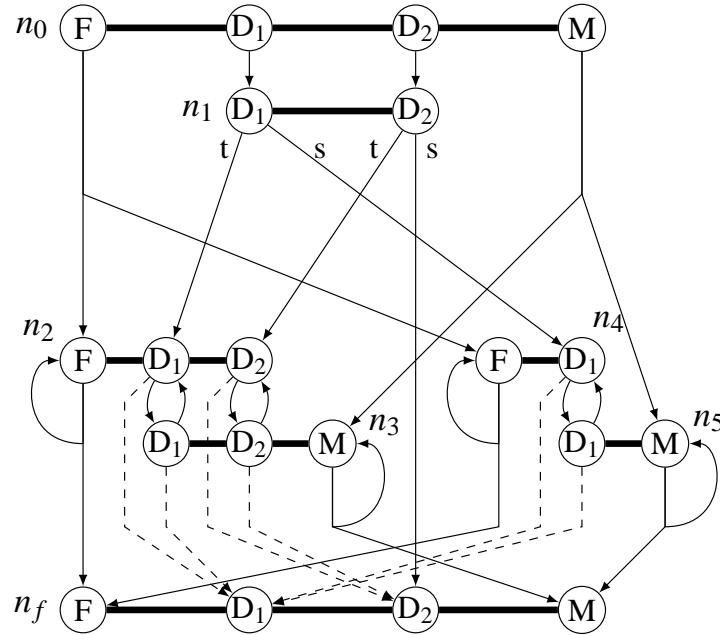


Figure 2: Atom control and determinism

Assume we want to model the following situation: In a family with Father (F), Mother (M) and two Daughters (D_1 and D_2), Daughter D_1 wants to go to a party. She can talk to each parent individually, but can choose beforehand whether to take her sister D_2 with her or not. Figure 2 models this negotiation. The solid edges for the daughters between n_2 and n_3 and between n_4 and n_5 “ask the other parent” outcomes, while the dashed edges represent the “yes” and “no” outcomes. Assume the daughters work together to reach termination. Then in $N_1 = \{n_1, n_2, n_3\}$ the daughters have a majority which we will interpret as “they can choose which outcome is taken”. Can the daughters force termination?

At atom n_1 the daughters decide whether D_2 should participate in the conversation with the parents (outcome t) or not (outcome s). If the daughters choose outcome s , then Father and Mother can force an infinite loop between n_4 and n_5 . On the contrary, if the daughters choose to stay together, then, since they control atom n_2 , they can force a “yes” or “no” outcome, and therefore termination.

The questions whether a coalition \mathcal{C} of agents can force termination or a certain outcome are special instances of the termination and concluding-outcome problems. In these instances, an atom n belongs to N_1 —the set of atoms controlled by Player 1—iff a strict majority of the agents of n are members of \mathcal{C} .

3.1 Coalitions

Before we turn to the termination and concluding-outcome problems, we briefly study coalitions. Intuitively, a coalition controls all the atoms where it has strict majority. We show that while the definition of the partition of the atoms N according to the participating agents may seem restricting, this is not the case: In all cases but the deterministic sound case, any partition can be reached, possibly by adding agents.

We define the partition of N via a partition of the agents: Let the agents A be partitioned into two sets A_1 and A_2 . Define $N_1 = \{n \in N : |P_n \cap A_1| > |P_n \cap A_2|\}$, $N_2 = N \setminus N_1$. Note that ties are controlled by A_2 .

We first show that in the nondeterministic and weakly deterministic case, this definition is equivalent to one where we decide control for each atom and not for each agent.

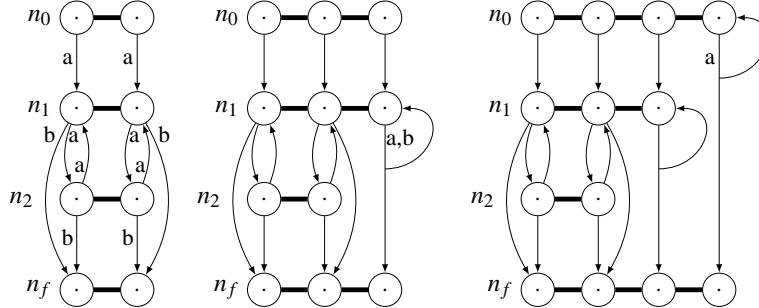


Figure 3: Atom control via additional agents

Consider the example given in Figure 3. On the left a deterministic negotiation with two agents is given. Assume the coalitions are $A_1 = \{A\}$ and $A_2 = \{B\}$. By the definition above, $N_2 = N$, thus coalition A_2 controls every atom. We want to change control of n_1 so that A_1 controls n_1 , changing the negotiation to a weakly deterministic one on the way. We add an additional agent a that participates in n_0, n_1, n_f as shown in Figure 3 in the middle and set $A_1 = \{A, a\}$. Now A_1 controls n_1 but also n_0 and n_f . We therefore add another agent b that participates in n_0, n_f as shown in Figure 3 on the right. Now the partition of atoms is exactly $N_1 = \{n_1\}$ and $N_2 = \{n_0, n_2, n_f\}$, as desired. In general, by adding nondeterministic agents to the negotiation, we can change the control for each atom individually. For each atom n whose control we wish to change, we add a number of agents to that atom, the initial atom n_0 and final atom n_f . We add nondeterministic edges for these agent from n_0 to $\{n, n_f\}$ for each outcome of n_0 and from n to $\{n, n_f\}$ for each outcome of n . It may be necessary to add more agents to n_0 that move to n_f in order to preserve the control of n_0 or n_f . This procedure changes the control of n while preserving soundness and weak determinism.

We proceed by showing that in the deterministic case, we cannot generate any atom control by adding more agents.

Lemma 3.3. *We cannot add deterministic agents to the negotiation on the left of Figure 3 in a manner, such that soundness is preserved and Player 1 controls n_1 , Player 2 controls n_2 .*

Proof. Consider again the deterministic negotiation game on the left of Figure 3. Assume we have added deterministic agents such that Player 1 controls n_1 . After the occurring sequence $\mathbf{x}_0 \xrightarrow{(n_0, a)} \mathbf{x}_1 \xrightarrow{(n_1, a)} \mathbf{x}_2$, those additional agents have moved deterministically, either to n_1 or n_f .¹

If any agent remains in n_1 , choosing outcome b in n_2 leads to a deadlock, otherwise, choosing a leads to a deadlock. Thus the negotiation is no longer sound. \square

¹Moving to n_2 would change the control there, thus additional agents have to be added to n_2 , we then can use a similar argument as follows by exchanging the roles of n_1 and n_2 .

4 The Termination Problem

We turn to the general complexity of the problem. It is easy to see that the termination problem can be solved in exponential time.

Theorem 4.1. *The termination problem is in EXPTIME.*

Proof. Sketch. (See the appendix for details.) We construct a concurrent reachability game on a graph such that Player 1 wins the negotiation game iff she wins this new game. The game has single exponential size in the size of the negotiation arena.

The nodes of the graph are either markings \mathbf{x} of the negotiation, or pairs $(\mathbf{x}, N_{\mathbf{x}})$, where \mathbf{x} is a marking and $N_{\mathbf{x}}$ is an independent set of atoms enabled at \mathbf{x} . Nodes \mathbf{x} belong to Scheduler, who chooses a set $N_{\mathbf{x}}$, after which the play moves to $(\mathbf{x}, N_{\mathbf{x}})$. At nodes $(\mathbf{x}, N_{\mathbf{x}})$ Players 1 and 2 concurrently select outcomes for their atoms in $N_{\mathbf{x}}$, and depending on their choice the play moves to a new marking. Player 1 wins if the play reaches the final marking \mathbf{x}_f . Since the winner of a concurrent game with reachability objectives played on a graph can be determined in polynomial time (see e.g. [2]), the result follows. \square

Unfortunately, there is a matching lower bound.

Theorem 4.2. *The termination problem is EXPTIME-hard even for negotiations in which every reachable marking enables at most one atom.*

Proof. Sketch. (See the appendix for details.) The proof is by reduction from the acceptance problem for alternating, linearly-bounded Turing machines (TM)[4]. Let M be such a TM with transition relation δ , and let x be an input of length n .

We define a negotiation with an agent I modeling the internal state of M , an agent P modeling the position of the head, and one agent C_i for each $1 \leq i \leq n$ modeling the i -th cell of the tape. The set of atoms contains one atom $n_{q,\alpha,k}$ for each triple (q, α, k) , where q is a state of M , k is the current position of the head, and α is the current symbol in the k -th tape cell.

The parties of the atom $n_{q,\alpha,k}$ are I , P and C_k , in particular, I and P are agents of all atoms. The atom $n_{q,\alpha,k}$ has one outcome \mathbf{r}_τ for each element $\tau \in \delta(q, \alpha)$, where τ is a triple consisting of a new state, a new tape symbol, and a direction for the head. We informally define the function $\mathcal{X}(n_{q,\alpha,k}, C_k, \mathbf{r}_\tau)$ by means of an example. Assume that, for instance, $\tau = (q', \beta, R)$, i.e., at control state q and with the head reading α , the machine can go to control state q' , write β , and move the head to the right. Then we have: (i) $\mathcal{X}(n_{q,\alpha,k}, I, \mathbf{r}_\tau)$ contains all atoms of the form $n_{q', _, _}$ (where $_$ stands for a wildcard); (ii) $\mathcal{X}(n_{q,\alpha,k}, P, \mathbf{r}_\tau)$ contains the atoms $n_{_, \beta, k+1}$; and (iii) $\mathcal{X}(n_{q,\alpha,k}, C_k, \mathbf{r}_\tau)$ contains all atoms $n_{_, \beta, _}$. If atom $n_{q,\alpha,k}$ is the only one enabled and the outcome \mathbf{r}_τ occurs, then clearly in the new marking the only atom enabled is $n_{q', \beta, k+1}$. So every reachable marking enables at most one atom.

Finally, the negotiation also has an initial atom that, loosely speaking, takes care of modeling the initial configuration.

The partition of the atoms is: an atom $n_{q,\alpha,k}$ belongs to N_1 if q is an existential state of M , and to N_2 if it is universal. It is easy to see that M accepts x iff Player 1 has a winning strategy. \square

Notice that if no reachable marking enables two or more atoms, Scheduler never has any choice. Therefore, the termination problem is EXPTIME-hard even if the strategy for Scheduler is fixed.

A look at points (i)-(iii) in the proof sketch of this theorem shows that the negotiations obtained by the reduction are highly nondeterministic. In principle we could expect a lower complexity in the deterministic case. However, this is not the case.

Theorem 4.3. *The termination problem is EXPTIME-hard even for deterministic negotiations in which every reachable marking enables at most one atom.*

Proof. Sketch. (See the appendix for details.) We modify the construction of Theorem 4.2 so that it yields a deterministic negotiation. The old construction has an atom $n_{q,\alpha,k}$ for each state q , tape symbol α , and cell index k , with I , P , and C_k as parties.

The new construction adds atoms $n_{q,k}$, with I and P as parties, and $n_{\alpha,k}$, with P and C_k as parties. Atoms $n_{q,k}$ have an outcome for each tape symbol α , and atoms $n_{\alpha,k}$ have an outcome for each state q . New atoms are controlled by Player 1.

In the new construction, after the outcome of $n_{q,\alpha,k}$ for transition $(q', \beta, R) \in \delta(q, \alpha)$, agent C_k moves to $n_{\beta,k}$, and agents I and P move to $n_{q',k+1}$. Intuitively, C_k waits for the head to return to cell k , while agents I and P proceed. Atom $n_{q',k+1}$ has an outcome for every tape symbol γ . Intuitively, at this atom Player 1 guesses the current tape symbol in cell $k+1$; the winning strategy corresponds to guessing right. After guessing, say, symbol γ , agent I moves directly to $n_{q',\gamma,k+1}$, while P moves to $n_{\gamma,k+1}$. Atom $n_{\gamma,k+1}$ has one outcome for every state of M . Intuitively, Player 1 now guesses the current control state q' , after which both P and C_{k+1} move to $n_{q',\gamma,k+1}$. Notice that all moves are now deterministic.

If Player 1 follows the winning strategy, then the plays mimic those of the old construction: a step like $(n_{q,\alpha,k}, (q', \beta, R))$ in the old construction, played when the current symbol in cell $k+1$ is γ , is mimicked by a sequence $(n_{q,\alpha,k}, (q', \beta, R)) (n_{q',k+1}, \gamma) (n_{\gamma,k+1}, q')$ of moves in the new construction. \square

5 Termination in Sound Deterministic Negotiations

In [5, 6] it was shown that the soundness problem (deciding whether a negotiation is sound) can be solved in polynomial time for deterministic negotiations and acyclic, weakly deterministic negotiations (the case of cyclic weakly deterministic negotiations is open), while the problem is PSPACE-complete for arbitrary negotiations. Apparently, Theorem 4.3 proves that the tractability of deterministic negotiations stops at game problems. We show that this is not the case. Well-designed negotiations are sound, since otherwise they contain atoms that can never occur (and can therefore be removed), or a deadlock is reachable. Therefore, we are only interested in the termination problem for sound negotiations. We prove that for sound deterministic negotiations (in fact, even for the larger class of weakly deterministic type 2 negotiations) the termination and concluding-outcome problems are solvable in polynomial time. For this, we show that the well-known attractor construction for reachability games played on graphs as arenas can be “lifted” to sound and weakly deterministic type 2 negotiation arenas.

Definition 5.1. *Let \mathcal{N} be a negotiation arena with a set of atoms $N = N_1 \cup N_2$. Given $n \in N$, let $P_{n,det}$ be the set of deterministic agents participating in n .*

The attractor of the final atom n_f is $\mathcal{A} = \bigcup_{k=0}^{\infty} \mathcal{A}_k$, where $\mathcal{A}_0 = \{n_f\}$ and

$$\begin{aligned} \mathcal{A}_{k+1} = \mathcal{A}_k \cup & \{n \in N_1 : \exists \mathbf{r} \in R_n \forall a \in P_{n,det} : \mathcal{X}(n, a, \mathbf{r}) \in \mathcal{A}_k\} \\ & \cup \{n \in N_2 : \forall \mathbf{r} \in R_n \forall a \in P_{n,det} : \mathcal{X}(n, a, \mathbf{r}) \in \mathcal{A}_k\} \end{aligned}$$

Given a marking $\mathbf{x} \neq \mathbf{x}_f$ of \mathcal{N} and a deterministic agent a , the attractor position of a at \mathbf{x} is the smallest k such that $\mathbf{x}(a) \in \mathcal{A}_k$, or ∞ if $\mathbf{x}(a) \notin \mathcal{A}$. Let a_1, \dots, a_k be the deterministic agents of \mathcal{N} . The (attractor) position vector of \mathbf{x} is the tuple (p_1, \dots, p_k) where p_i is the attractor position of a_i .

Theorem 5.2. *Let \mathcal{N} be a sound, weakly deterministic type 2 negotiation arena. Player 1 has a winning strategy in the termination game iff $n_0 \in \mathcal{A}$.*

Proof. We start with an observation: If all deterministic agents are ready to take part in n_f , then n_f and only n_f is enabled. Indeed, since deterministic agents are only ready to engage in at most one atom, the only atom with a deterministic party that can be enabled is n_f . Moreover, by weak determinism type 2 every atom has a deterministic party, and so no atom other than n_f can be enabled. Finally, by soundness at least one atom is enabled, and so n_f is the only enabled atom.

(\Leftarrow): Assume that $n_0 \in \mathcal{A}$. We fix the *attractor strategy* for Player 1. We define the attractor index of an atom $n \in \mathcal{A}$ as the smallest k such that $n \in \mathcal{A}_k$. The strategy for an atom $n \in N_1 \cap \mathcal{A}$ is to choose any outcome such that all deterministic parties of n move to an atom of smaller attractor index; formally, we choose any outcome r such that for every deterministic party a the singular atom in $\mathcal{X}(n, a, r)$ has smaller attractor index than n . Such an outcome exists by construction of \mathcal{A} . For atoms $n \in N_1 \setminus \mathcal{A}$ we choose an arbitrary atom. Notice that this strategy is not only memoryless, but also independent of the current marking.

We show that the attractor strategy is winning. By the definition of the game, we have to prove that every play following the strategy ends with n_f occurring. By the observation above, it suffices to prove that the play reaches a marking at which every deterministic agent is ready to engage in n_f .

Assume there is a play π where Player 1 plays according to the attractor strategy, which never reaches such a marking. Then the play never reaches the final marking \mathbf{x}_f either. We claim that at all markings reached along π , the deterministic agents are only ready to engage in atoms of \mathcal{A} . We first observe that, initially, all deterministic agents are ready to engage in n_0 , and $n_0 \in \mathcal{A}$. Now, assume that in some marking reached along π the deterministic agents are only ready to engage in atoms of \mathcal{A} . Then, by weak determinism type 2, all enabled atoms belong to \mathcal{A} , and therefore also the atoms chosen by Scheduler. By the definition of \mathcal{A} , after an atom of $N_2 \cap \mathcal{A}$ occurs, the deterministic agents are ready to engage in atoms of \mathcal{A} only; by the definition of the attractor strategy, the same holds for atoms of $N_1 \cap \mathcal{A}$. This concludes the proof of the claim.

Since all markings \mathbf{x} reached along π satisfy $\mathbf{x} \neq \mathbf{x}_f$ and $\mathbf{x}(a) \in \mathcal{A}$ for every deterministic agent a , they all have an associated attractor position vector whose components are natural numbers. Let P_k denote the position vector of the marking reached after $k \geq 0$ steps in π . Initially only the initial atom n_0 is enabled, and so $P_0 = (k_0, k_0, \dots, k_0)$, where k_0 is the attractor position of n_0 . We have $k_0 < |N|$, the number of atoms of the negotiation arena \mathcal{N} . Given two position vectors $P = (p_1, \dots, p_k)$ and $P' = (p'_1, \dots, p'_k)$, we say $P \prec P'$ if $p_i \leq p'_i$ for every $1 \leq i \leq k$, and $p_i < p'_i$ for at least one $1 \leq i \leq k$. By the definition of the attractor strategy, the sequence P_0, P_1, \dots of attractor positions satisfies $P_{i+1} \prec P_i$ for every i . Since \prec is a well-founded order, the sequence is finite, i.e., the game terminates. By the definition of the game, it terminates at a marking that does not enable any atom. Since, by assumption, the play never reaches the final marking \mathbf{x}_f , this marking is a deadlock, which contradicts the soundness of \mathcal{N} .

(\Rightarrow): As this part is dual to part \Leftarrow , we only sketch the idea. The complete proof can be found in the appendix.

Let $\mathcal{B} = N \setminus \mathcal{A}$, and assume $n_0 \in \mathcal{B}$. A winning strategy for Player 2 is to choose for an atom $n \in N_2 \cap \mathcal{B}$ any outcome r such that at least one deterministic agent moves to an atom not in \mathcal{A} . Such an outcome exists by construction of \mathcal{A} . For atoms in $N_2 \setminus \mathcal{B}$ we chose an arbitrary outcome.

This strategy achieves the following invariant: If at some marking \mathbf{x} reached along a play according to this strategy, there is a deterministic agent a that satisfies $\mathbf{x}(a) = n$ and $n \in \mathcal{B}$, then the same holds after one more step of the play. We then conclude that since $n_0 \in \mathcal{B}$ this invariant holds in every step of the play. Therefore n_f , in which all deterministic agents participate and which is not in \mathcal{B} , can never be

enabled. Finally, because of soundness a play cannot end in a deadlock and thus every play will be of infinite length. \square

Corollary 5.3. *For the termination game over sound and weakly deterministic type 2 negotiations, the following holds:*

- (a) *The game collapses to a two-player game.*
- (b) *Memoryless strategies suffice for both players.*
- (c) *The winner and the winning strategy can be computed in $O(|R| * |A|)$ time, where A is the set of agents and $|R|$ the total number of outcomes of the negotiation.*

Proof. (a) and (b): The attractor computation and the strategies used in the proof of Theorem 5.2 are independent of the choices of Scheduler; the strategies are memoryless.

(c) An algorithm achieving this complexity can be found in the appendix. \square

We still have to consider the possibility that requiring soundness alone, without the addition of weak determinism type 2, already reduces the complexity of the termination problem. The following theorem shows that this is not the case, and concludes our study. The proof is again an adaption of the reduction from turing machines and can be found in the appendix.

Theorem 5.4. *The termination problem is EXPTIME-hard for sound negotiation arenas.*

6 The Concluding-Outcome Problem

We demonstrate that the algorithm can also be used to solve the concluding-outcome problem. Remember that for each agent a we have selected a set G_a of outcomes that lead to the final atom and that we want occur. The key idea is to modify the negotiation in the following way: We add two atoms per agent, say $good_a$ and bad_a , and redirect the outcomes of that agent that lead to the final atom to $good_a$ if they are in G_a and to bad_a otherwise. We illustrate this approach by example.

We slightly change the setting of Father, Mother and two Daughters such that both daughters want to go to the party. In the negotiation, that results in two additional atoms where D_2 talks alone with each parent. Figure 4 shows the new negotiation omitting all edges of father and mother to simplify the representation. They both have a nondeterministic edge from each of their ports leading to all of their respective ports.

Imagine the goal of Daughter D_1 is to get a “yes” answer for herself, but a “no” answer for D_2 , who always spoils the fun. Will a coalition with one parent suffice to achieve the goal? To answer this question, we modify the negotiation before applying the construction as shown in Figure 4. We introduce dummy atoms $good_i$ and bad_i for each Daughter i , and we redirect “yes” transitions leading to n_f to $good_1$ and bad_2 (these transitions are represented as dashed lines), analogously we redirect “no” transitions to bad_1 and $good_2$ (represented as dotted lines).

We apply the algorithm with a slight alteration: Instead of starting from n_f , we initially add all newly introduced good atoms to the attractor. Applying the algorithm for the coalition Father- D_1 yields $\{good_1, bad_2, n_4\}$ as attractor; for Mother- D_1 we get $\{good_1, bad_2, n_5\}$. So neither parent has enough influence to achieve the desired outcome.

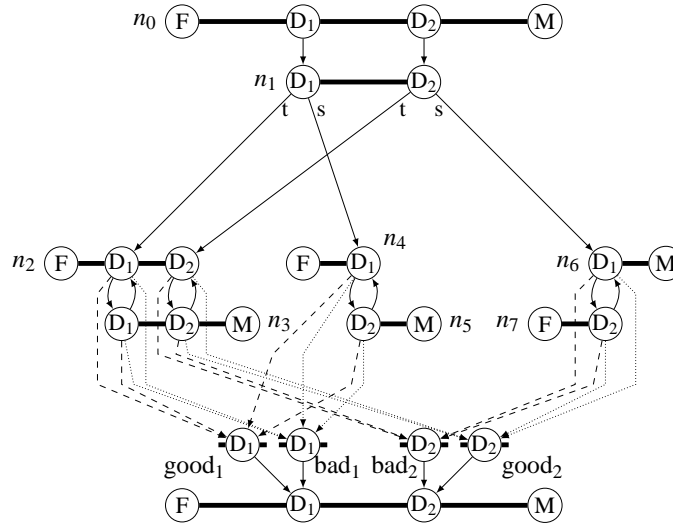


Figure 4: Applying the algorithm to the final-outcome problem. Father and mother edges omitted.

7 Conclusions and Related Work

We have started the study of games in the negotiation model introduced in [5, 6]. Our results confirm the low computational complexity of deterministic negotiations, however with an important twist: while even the simplest games are EXPTIME-hard for arbitrary deterministic negotiations, they become polynomial in the *sound* case. So soundness, a necessary feature of a well designed negotiation, also turns out to have a drastic beneficial effect on the complexity of the games.

We have shown that our games are also polynomial for sound and weakly deterministic negotiations. However, the complexity of deciding soundness for this case is unknown. We conjecture that it is also polynomial, as for the deterministic case.

The objectives we have considered so far are qualitative: Either a coalition can reach termination or not, either it can enforce a concluding outcome or not. A possibility for future studies would be to look at quantitative objectives.

We have only considered 2-player games, since in our settings the behavior of the third player (the Scheduler) is either irrelevant, or is controlled by one of the other two players. We intend to study the extension to a proper 3-player game, or to a multi player game. Combined with qualitative objectives, this allows for multiple interesting questions: Which coalition of three agents can reach the best payoff? Which two agents should a given agent side with to maximize his payoff? Are there “stable” coalitions, that means, no agents can change to the opposing coalition and improve his payoff?

References

- [1] Luca de Alfaro, Thomas A. Henzinger & Orna Kupferman (2007): *Concurrent reachability games*. *Theor. Comput. Sci.* 386(3), pp. 188–217. Available at <http://dx.doi.org/10.1016/j.tcs.2007.07.008>.
- [2] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *J. ACM* 49(5), pp. 672–713. Available at <http://doi.acm.org/10.1145/585265.585270>.
- [3] Tomás Brázdil, Petr Jancar & Antonín Kucera (2010): *Reachability Games on Extended Vector Addition Systems with States*. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide

- & Paul G. Spirakis, editors: *ICALP (2), Lecture Notes in Computer Science* 6199, Springer, pp. 478–489. Available at http://dx.doi.org/10.1007/978-3-642-14162-1_40.
- [4] Ashok K. Chandra, Dexter Kozen & Larry J. Stockmeyer (1981): *Alternation*. *J. ACM* 28(1), pp. 114–133, doi:10.1145/322234.322243.
 - [5] Javier Esparza & Jörg Desel (2013): *On Negotiation as Concurrency Primitive*. In Pedro R. D’Argenio & Hernán C. Melgratti, editors: *CONCUR, Lecture Notes in Computer Science* 8052, Springer, pp. 440–454. Available at http://dx.doi.org/10.1007/978-3-642-40184-8_31.
 - [6] Javier Esparza & Jörg Desel (2014): *On Negotiation as Concurrency Primitive II: Deterministic Cyclic Negotiations*. In Anca Muscholl, editor: *FoSSaCS, Lecture Notes in Computer Science* 8412, Springer, pp. 258–273. Available at http://dx.doi.org/10.1007/978-3-642-54830-7_17.
 - [7] Paul Gastin, Nathalie Sznajder & Marc Zeitoun (2009): *Distributed synthesis for well-connected architectures*. *Formal Methods in System Design* 34(3), pp. 215–237. Available at <http://dx.doi.org/10.1007/s10703-008-0064-7>.
 - [8] Blaise Genest, Hugo Gimbert, Anca Muscholl & Igor Walukiewicz (2013): *Asynchronous Games over Tree Architectures*. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska & David Peleg, editors: *ICALP (2), Lecture Notes in Computer Science* 7966, Springer, pp. 275–286. Available at http://dx.doi.org/10.1007/978-3-642-39212-2_26.
 - [9] Antonín Kucera (2012): *Playing Games with Counter Automata*. In Alain Finkel, Jérôme Leroux & Igor Potapov, editors: *RP, Lecture Notes in Computer Science* 7550, Springer, pp. 29–41. Available at http://dx.doi.org/10.1007/978-3-642-33512-9_4.
 - [10] Swarup Mohalik & Igor Walukiewicz (2003): *Distributed Games*. In Paritosh K. Pandya & Jaikumar Radhakrishnan, editors: *FSTTCS, Lecture Notes in Computer Science* 2914, Springer, pp. 338–351. Available at http://dx.doi.org/10.1007/978-3-540-24597-1_29.
 - [11] Ariel Rubinstein (1982): *Perfect equilibrium in a bargaining model*. *Econometrica* 50(1), pp. 97–109, doi:10.2307/1912531.
 - [12] Igor Walukiewicz (2001): *Pushdown Processes: Games and Model-Checking*. *Inf. Comput.* 164(2), pp. 234–263. Available at <http://dx.doi.org/10.1006/inco.2000.2894>.